AMENDMENTS TO THE CLAIMS:

This listing of claims replaces all prior versions and listings of claims in the application:

LISTING OF CLAIMS:

1. (Previously Presented) A processor comprising:

a first multi-threaded processor engine configured for connection to a serial link;

a second multi-threaded processor engine, coupled to the first multi-threaded processor engine by an interface, to process data received by the first multi-threaded processor over the serial link and to provide the processed data to the first multi-threaded processor engine for transmission over the serial link; and

one or more communication data structures usable by the first and second multi-threaded processor engines to control interaction therebetween,

wherein at least one of the one or more communication data structures is usable by the first and second multi-threaded processor engines to control the rate at which the processed data is provided to the first multi-threaded processor engine by the second multi-threaded engine.

2. (Cancelled)

3. (Previously Presented) The processor of claim 1 wherein the data comprise a frame-based protocol.

4. (Previously Presented) The processor of claim 3 wherein the frame-based protocol comprises High Data Link Control (HDLC).

5. (Previously Presented) The processor of claim 1 wherein the data comprise a cell-based protocol.

6. (Previously Presented) The processor of claim 5 wherein the cell-based protocol comprises ATM.

7. (Previously Presented) The processor of claim 6 wherein the cell-based protocol comprises Inverse Multiplexing for ATM.

8. (Previously Presented) The processor of claim 1 wherein the data comprises frame-based and cell-based protocols.

9. (Currently Amended) The processor of claim [[2]] 1 wherein the second multi-threaded processor engine is capable of operating at a faster rate than the first multi-threaded processor engine.

10. (Previously Presented) The processor of claim 9 wherein the at least one of the one or more communication data structures includes a flow control ring of entries written by the first multi-threaded processor engine and read by the second multi-threaded processor engine.

11. (Previously Presented) The processor of claim 10 wherein each flow control ring entry comprises a flow control message that includes a channel number identifying a channel corresponding to one of multiple channels supported by the serial link.

12. (Previously Presented) The processor of claim 11 wherein the second multi-threaded processor engine is operable to use the channel number to determine a flow control amount of data to be provided to the first multi-threaded processor engine for the channel identified by the channel number.

13. (Previously Presented) The processor of claim 12 wherein the second multi-threaded processor engine is operable to poll the flow control ring, and wherein the flow control amount of data and the frequency with which the flow control ring is polled are sufficient to prevent an underflow of data to the channel for transmission.

14. (Previously Presented) The processor of claim 12 wherein the second multi-threaded processor engine is operable to poll the flow control ring, and wherein the flow control amount

of data and the frequency with which the flow control ring is polled are sufficient to prevent a critical underflow of data to the channel for transmission.

15. (Previously Presented) The processor of claim 9 wherein the at least one or more communication data structures includes a shared memory comprising memory locations corresponding to each of the channels that handles cell data, each memory location storing a count value indicating the number of cells transmitted by the first multi-threaded processor engine for the channel to which such memory location corresponds.

16. (Previously Presented) The processor of claim 15 wherein the second multi-threaded processor engine is operable to poll the memory locations.

17. (Previously Presented) The processor of claim 16 wherein the second multi-threaded processor engine, after polling one of the memory locations of a selected one of the channels, is operable to compare the count value with another count value indicative of a number of cells scheduled for transmission, maintained by the second multi-threaded processor engine, to determine a number of cells in flight.

18. (Previously Presented) The processor of claim 1 wherein the multi-threaded second processor engine is operable to compare the number of cells in flight to a predetermined threshold to determine if flow control is necessary to prevent an overflow condition from

occurring for the channel.

19. (Previously Presented) The processor of claim 1 wherein the interface comprises: receive buffers written to by the first multi-threaded processor engine and read by the second multi-threaded processor engine; transmit buffers written to by the second multi-threaded processor engine and read by the first multi-threaded processor engine; and wherein the receive and transmit buffers comprise buffer elements configured to store a software prepend header followed by a programmed number of bytes.

20. (Previously Presented) The processor of claim 19 wherein the first multi-threaded processor engine is configured to operate in a frame-base mode to receive and reassemble the data when the data comprises a frame-based protocol, and is further configured to operate in a cell-based mode to receive and reassemble the data when the data comprises a cell-based protocol.

21. (Previously Presented) The processor of claim 20 wherein one of the receive buffers is designated for use by the first multi-threaded processor engine, and wherein the first multi-threaded processor engine is operable to receive and process data until an entire data unit, comprising either a cell in the cell-based mode or up to a programmed number of bytes of frame data in the frame-based mode, has been received and reassembled, at which point the first multi-threaded processor engine appends to the reassembled data the software prepend header and

writes the software prepend header and the reassembled data to an allocated one of the buffer elements in the designated receive buffer.

22. (Previously Presented) The processor of claim 21 wherein the second multi-thread processor engine is operable to use the software prepend header to further process the reassembled data.

23. (Previously Presented) The processor of claim 22 wherein the software prepend header includes an indicator of whether the reassembled data comprises cell data or frame data.

24. (Previously Presented) The processor of claim 1 wherein the first multi-threaded processor engine is configured to operate as a co-processor for the second multi-threaded processor.

25. (Previously Presented) The processor of claim 24 wherein the one or more communication data structures includes at least one communication ring to pass first messages from the second multi-threaded processor engine to the first multi-threaded processor engine to describe a location of data to be processed and type of processing to be performed by the first multi-threaded processor engine, and to pass second messages from the first multi-threaded processor engine to the second multi-threaded processor engine to describe a location of the data after processing.

26. (Previously Presented) The processor of claim 1 wherein the serial link can be any one of T1, E1 and J1 links.

27. (Previously Presented) A processor comprising:

a first multi-threaded processor engine;

a second multi-threaded processor engine operable to process data received from a network via a network interface; and

at least one or more communication data structures,

wherein the first multi-threaded processor engine is configured to operate as a co-processor for the second multi-threaded processor engine,

wherein the at least one or more communication data structures is usable by the first and second multi-threaded processor engines to control the rate at which the processed data is provided to the first multi-threaded processor engine by the second multi-threaded engine,

wherein the at least one or more communication data structures includes a shared memory comprising memory locations corresponding to each of the channels that handles cell data, each memory location storing a count value indicating the number of cells transmitted by the first multi-threaded processor engine for the channel to which such memory location corresponds,

wherein the second multi-threaded processor engine, after polling one of the memory locations of a selected one of the channels, is operable to compare the count value with another

count value indicative of a number of cells scheduled for transmission, maintained by the second multi-threaded processor engine, to determine a number of cells in flight.

28. (Previously Presented) The processor of claim 27 wherein the network interface comprises the first multi-threaded processor engine.

29. (Previously Presented) The processor of claim 27 wherein the network interface comprises an external media device.

30. (Previously Presented) The processor of claim 27 wherein the at least one or more communication data structures comprises:

a first communication data structure usable to pass first messages from the second multi-threaded processor engine to the first multi-threaded processor engine to describe a location of data to be processed and type of processing to be performed by the first multi-threaded processor engine as a co-processor; and

a second communication data structure to pass second messages from the first multi-threaded processor engine to the second multi-threaded processor engine to describe a location of the data after processing by the first multi-threaded processor engine.

31. (Previously Presented) The processor of claim 27 wherein the type of processing to be performed by the first multi-threaded processor engine as a co-processor includes at least one cryptographic function.

32. (Previously Presented) The processor of claim 27 wherein the type of processing to be performed by the first multi-threaded processor engine as a co-processor includes authentication.

33. (Previously Presented) The processor of claim 27 wherein the type of processing to be performed by the first multi-threaded processor engine as a co-processor includes a hashing function.

34. (Previously Presented) A network device comprising:

a line card to couple to one or more data links, the line card comprising a network processor that comprises:

a first multi-threaded processor engine;

a second multi-threaded processor engine, coupled to the first multi-threaded processor engine by an interface, configured to process data received over the one or more data links via a network device or the first multi-threaded processor engine when such processor is configured for use as a physical layer device to receive and transmit serial data over at least one of the one or more data links, and to provide the processed data to the first multi-threaded processor engine

for transmission if the data that was processed was received from the first multi-threaded processor engine; and

one or more communication data structures usable by the first and second multi-threaded processor engines to control interaction therebetween,

wherein at least one of the one or more communication data structures is usable by the first and second multi-threaded processor engines to control the rate at which the processed data is provided to the first multi-threaded processor engine by the second multi-threaded engine,

wherein the at least one or more communication data structures includes a shared memory comprising memory locations corresponding to each of the channels that handles cell data, each memory location storing a count value indicating the number of cells transmitted by the first multi-threaded processor engine for the channel to which such memory location corresponds,

wherein the second multi-threaded processor engine, after polling one of the memory locations of a selected one of the channels, is operable to compare the count value with another count value indicative of a number of cells scheduled for transmission, maintained by the second multi-threaded processor engine, to determine a number of cells in flight.

35. (Original) The network device of claim 34 wherein at least one of the one or more communication data structures is usable by the first and second multi-threaded processor engines to control the rate at which the processed data is provided to the first multi-threaded processor engine by the second microprocessor engine.

36. (Original) The network device of claim 34 wherein the first multi-threaded processor engine is configured to operate as a co-processor for the second multi-threaded processor during the data processing.

37. (Original) The network device of claim 36, further comprising: a first communication data structure usable to pass first messages from the second multi-threaded processor engine to the first multi-threaded processor engine to describe a location of data to be processed and type of processing to be performed by the first multi-threaded processor engine as a co-processor; and a second communication data structure to pass second messages from the first multi-threaded processor engine to the second multi-threaded processor engine to describe a location of the data after processing by the first multi-threaded processor engine.

38. (Previously Presented) A method comprising:

providing a first multi-threaded processor engine configured for connection to a serial link;

providing a second multi-threaded processor engine, coupled to the first multi-threaded processor engine by an interface, to process data received by the first multi-threaded processor over the serial link and to provide the processed data to the first multi-threaded processor engine for transmission over the serial link; and

enabling use of one or more communication data structures by the first and second multi-threaded processor engines to control interaction therebetween,

wherein at least one of the one or more communication data structures is usable by the first and second multi-threaded processor engines to control the rate at which the processed data is provided to the first multi-threaded processor engine by the second multi-threaded engine,

wherein the at least one or more communication data structures includes a shared memory comprising memory locations corresponding to each of the channels that handles cell data, each memory location storing a count value indicating the number of cells transmitted by the first multi-threaded processor engine for the channel to which such memory location corresponds,

wherein the second multi-threaded processor engine, after polling one of the memory locations of a selected one of the channels, is operable to compare the count value with another count value indicative of a number of cells scheduled for transmission, maintained by the second multi-threaded processor engine, to determine a number of cells in flight.

39. (Original) The method of claim 38 wherein enabling comprises:

enabling use of at least one of the one or more communication data structures by the first and second multi-threaded processor engines to control the rate at which the processed data is provided to the first multi-threaded processor engine by the second multi-threaded processor engine.

40. (Original) The method of claim 38 further comprising: enabling use of the first multi-threaded processor engine as a co-processor to the second multi-threaded processor engine.

41. (Original) The method of claim 40, wherein the communication data structures comprise: a first communication data structure usable to pass first messages from the second multi-threaded processor engine to the first multi-threaded processor engine to describe a location of data to be processed and type of processing to be performed by the first multi-threaded processor engine as a co-processor; and a second communication data structure to pass second messages from the first multi-threaded processor engine to the second multi-threaded processor engine to describe a location of the data after processing by the first multi-threaded processor engine.

42. (Previously Presented) A computer-readable medium having stored thereon instructions that when executed by a machine result in the following:

performing multi-threaded packet processing on data received by a multi-threaded physical layer processor engine over a serial link;

providing the processed data to the multi-threaded physical layer processor engine for transmission over the serial link; and

using one or more communication data structures to control interaction with the multi-threaded physical layer processor engine, at least one or more communication data structures comprising a shared memory comprising memory locations corresponding to each of the

channels that handles cell data, each memory location storing a count value indicating the number of cells transmitted by the multi-threaded physical layer processor engine for the channel to which such memory location corresponds,

polling one of the memory locations of a selected one of the channels; and

comparing the count value with another count value indicative of a number of cells scheduled for transmission to determine a number of cells in flight.

43. (Currently Amended) The computer-readable medium of claim 42 wherein the communication data structures are usable to control the rate at which the processed data is provided to the multi-threaded physical layer network processor engine.

44. (Previously Presented) A computer-readable medium having stored thereon instructions that when executed by a machine result in the following:

enabling multi-threaded packet processing of data received from a communications medium via a multi-threaded physical layer network processor engine, which receives serial data from a serial link, or other media device; and

using the multi-threaded physical layer network processor engine as a co-processor to perform a hardware accelerator task associated with the multi-threaded packed processing.

45. (Currently Amended) The computer-readable medium of claim 44 further comprising:

using one or more communication data structures to pass control messages to and receive control messages from the multi-threaded physical layer network processor engine operating in the co-processor mode.